

**PRUEBA N°1 ESTRUCTURA DE DATOS  
SEGUNDO SEMESTRE 2005**

NOMBRE:..... SECCIÓN:.....

1. Se dispone de un stack S que contiene los datos (menos de 1000) correspondientes a los postulantes al ciclo básico (1° a 8°) del colegio Nido de Tiuques Jai Eskul. Cada dato es un número entero de cuatro dígitos en la forma cPPP, donde c representa el curso al cual el alumno postula y PPP el puntaje obtenido en la prueba de selección. Los datos se pusieron en S según el orden de llegada de los postulantes. Implementar el operador **Mejores(S, V)** que, a partir de S, almacena en V (de tipo int Vector[8]) sólo los mejores puntajes obtenidos por los postulantes.

2. Algunas formas sintácticas en lenguaje C y sus equivalentes en lenguaje Ch, son:

if(e) s1 else s2 equivalente si(e, s1, s2)

{s1; s2; s3; sk; } equivalente {s1, s2, s3, sk, }

p + q equivalente mas(p, q)

p - q equivalentemenos(p, q)

p \* q equivalente por(p, q)

p \*\* q equivalente eleva(p, q)

int f(int p, int q) equivalente ent f(ent p, ent q)

p < q equivalente menor(p, q)

return(e) equivalente retorna(e)

Además, la representación estática, con exponentes implícitos, de un polinomio, se basa en los siguientes tipos de datos: de tipo((0..100, expo), (real [expo], polinomio))

Sin utilizar la sentencia de asignación, implementar en Ch:

a) El operador **valorterm(c, x, e)** que retorna el valor de  $cx^e$ .

b) El operador **recursivo valorpoli(p, n, x)** que evalúa el polinomio p (de grado n 100) en el punto x.

3. El TAD matriz dispersa se representa de la forma:

```
typedef struct Dispersa
{int f;
  int c;
  base nulo;
  Compacta a;
};
```

y, sobre una variable S de tipo Dispersa, es posible aplicar los siguiente operadores:

Eom(S) : Indica que en S no quedan elementos.

Primero(S, i, j) : Retorna los índices del elemento actual en S y se posiciona en el siguiente elemento.

Agregar(S, i, j, e) : Agrega en S el elemento e en la posición i, j.

Valor(S, i, j) : Retorna el valor del elemento de índices i, j en S.

Considérense las matrices A, B, C, de tipo Dispersa y A, B del mismo orden, para implementar el operador **Sumar(A, B, C)**, correspondiente a la operación  $C = A + B$ , tal que  $C_{ij} = A_{ij} + B_{ij}$  sí y sólo sí  $A_{ij} \neq 0$  y  $B_{ij} \neq 0$ .

PAUTA 1

```
void Mejores(Stack S, Vector V)
{ int e, c, punt;
  while (!Empty(S))
  {
    e = Pop(S);
    c = d/1000;
    punt = d % 1000;
    if (punt > V[c-1])
      V[c-1] = punt;
  }
}
```

PAUTA 2

```
a)  real valorterm (real c, real x, expo e)
{ retorna(por(c, eleva(x, e))) }

b)  real valorpoli (polinomio p, expo n, real x)
{ si (menor(n, 0), retorna(0), retorna(mas(valorterm(p[n], x, n), valorpoli(p, menos(n, 1), x)))) }
```

PAUTA 3

```
void Sumar(Dispersa A, Dispersa B, Dispersa C)
{ int i, j;
  Base e;
  C.f = A.f;
  C.c = A.c
  while(!Eom(A))
  { Primero(A, i, j);
    e = Valor(B, i, j);
    if(e != 0);
    Agregar(C, j, i, Valor(A, i, j) + e);
  }
}
```